CLAIMS

1. A method of communicating isochronous data from a source attached to a first serial bus to a sink attached to a second serial bus, said first and second buses operating cyclically with similar but unsynchronised cycle periods, the method comprising the steps of determining the tolerances with respect to frequency between the first and second bus cycle periods, assembling the data for transmission into packets, allocating to said packets a variable size data payload dependent on the tolerances, including within the packets a header indicating the size of the payload, receiving the packets at the data sink, and extracting the data payload from the packets using the packet header indication of the size of the data payload.

2. A method as claimed in Claim 1, comprising the steps of receiving data packets transmitted from the first bus at the second bus, entering the received packets into a received packets register, transferring each received packet into a first in first out (FIFO) memory when sufficient space exists in said FIFO memory, reading out output data packets from said FIFO memory at the cycle rate of the second bus, said output packets containing a data payload which is chosen to keep the average contents of the FIFO memory substantially constant.

3. A method as claimed in Claim 2, comprising the further step of including in the data payload a code indicating the end of the data contained within the received packet whose data payload forms the initial part of the data payload of the output data packet.

4. A method as claimed in Claim 3, comprising the steps of receiving output data packets from the second bus, reassembling them into data packets as received from the first bus, and transmitting the reassembled

packets to a third bus operating cyclically with similar cycle periods to the first and second buses but not being synchronised with either of the other buses.

5. A method as claimed in Claim 3, including the step of inserting into the data payload of the output data packet a code indicating the length of the data payload of data received in a received data packet subsequent to the received data packet whose data payload occupies the initial portion of the data payload of the output data packet.

6. A method as claimed in Claim 4, including the step of inserting into the data payload of the output data packet a code indicating the length of the data payload of data received in a received data packet subsequent to the received data packet whose data payload occupies the initial portion of the data payload of the output data packet.

7. A communications network for communicating isochronous data comprising first and second buses, one or more data sources connected to the first data bus, one or more data sinks connected to the second data bus for receiving isochronous data from the data source, each data bus having similar but unsynchronised cycle periods said communications network further comprising a first interface arrangement connected to said first bus, said first interface arrangement comprising a packet assembly arrangement which assembles at the cycle rate of the first bus data packets comprising a header portion and a data payload portion, the header portion including data defining the length of the payload portion, and an output from which the data packets can be transmitted to a second interface arrangement connected to said second bus, said second interface arrangement comprising a buffer memory arrangement which receives the data packets sent by the first interface arrangement, separates the data payload from the data packet, reassembles data packets at the cycle rate of the second bus in such a manner that the quantity of data in the buffer memory kept substantially constant by varying

the quantity of data in the payload of the data packet, and applies the reassembled data packets to the second bus.

8. A communications network as claimed in Claim 6, in which said second interface arrangement comprises means for inserting data, which defines the end of the data payload of the received data packet which occupies the first part of the data payload of the reassembled packet, into the reassembled data packet within the data payload of the reassembled data packet.

9. A communications network as claimed in Claim 7, in which said second interface arrangement further inserts data representing the length of the data payload of the next received packet.

10. A communication network as claimed in Claim 8, comprising a third interface arrangement arranged to receive data packets from said second bus for transfer to a third bus, said third interface arrangement comprising a packet received register for receiving data packets from the second bus, a FIFO into which the received packets are transferred when space is available for them, and a packet reassembler which uses the data indicating the end of received packets to reassemble for transfer to the third bus data packets identical to the data packets transmitted from the first bus.

11. A communication network as claimed in Claim 9, comprising a third interface arrangement arranged to receive data packets from said second bus for transfer to a third bus, said third interface arrangement comprising a packet received register for receiving data packets from the second bus, a FIFO into which the received packets are transferred when space is available for them, and a packet reassembler which uses the data indicating the end of received packets to reassemble for transfer to the third bus data packets identical to the data packets transmitted from the first bus.

12. A data bus arrangement including an input interface arrangement for receiving data packets from another data bus having a similar but unsynchronised cycle period, said input interface arrangement comprising a buffer memory arrangement which receives the data packets from the other bus, said data packets having a header portion including an indication of the data payload in the data packet, said interface arrangement being operative to separate the data payload from the data packet, reassemble data packets at the cycle rate of the bus in such a manner that the quantity of data in the buffer memory is kept substantially constant by varying the quantity of data in the payload of the data packet, and apply the reassembled data packets to the second bus.

13. A data bus arrangement as claimed in Claim 12, wherein the reassembled data packet includes data indicating the boundary, within the data payload of the reassembled packet, of the data payload in the received data packet which forms the first portion of the data payload of the reassembled packet.

14. A data bus arrangement as claimed in 13, wherein the reassembled data includes data indicating the size of the data payload of the received data packet that starts from the end of the data payload of the first received data packet.

15. A data bus arrangement as claimed in Claim 13, including an output interface arrangement for receiving data packets from the bus and transmitting them to a further data bus having a similar but unsynchronised cycle period, said output interface arrangement comprising a receive buffer memory in which the data payload from a received data packet is stored, an output FIFO into which the data payload is transferred when space is available, a packet reassembler which uses the data within the data payload indicating the boundary of the payload in the data packet received from the other bus to produce a data packet having the same data payload as that of

the data packet received from the other bus, and transmission means for transmitting the reassembled data packet to the further data bus.